



Improved quantum logic circuit for reversible quantum computation

Jeong Ryeol Choi^{1✉}, Ji Nny Song²

1. Professor, Department of Radiologic Technology, Daegu Health College, Yeongsong-ro 15, Buk-gu, Daegu 41453, Republic of Korea

2. Researcher, Department of Radiologic Technology, Daegu Health College, Yeongsong-ro 15, Buk-gu, Daegu 41453, Republic of Korea

✉ **Corresponding author:**

Department of Radiologic Technology,
Daegu Health College,
Yeongsong-ro 15, Buk-gu, Daegu 41453,
Republic of Korea,
E-mail: choiardor@hanmail.net

Article History

Received: 20 June 2017

Accepted: 22 July 2017

Published: 1 August 2017

Citation

Jeong Ryeol Choi, Ji Nny Song. Improved quantum logic circuit for reversible quantum computation. *Discovery*, 2017, 53(260), 448-453

Publication License



© The Author(s) 2017. Open Access. This article is licensed under a [Creative Commons Attribution License 4.0 \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).

General Note

Article is recommended to print as color digital version in recycled paper.

ABSTRACT

Although quantum computing which uses a quantum algorithm is a complex and fragile process, the recent critical advancement of its relevant technology may ultimately lead to the commercialization of quantum computers. Quantum computers have comparatively more powerful computing abilities than existing ones which operate on the basis of classical mechanics. Meanwhile,

reversible quantum computation is important in processing arithmetic operations using quantum computer systems. Recently, a quantum circuit for reversible quantum computation based on Excess-3 code has been proposed by one of us. In the present work, we improve this circuit by reducing garbage operation lines. While we have confirmed by tests that this improvement gives exact results for computation, it may provide more efficient quantum computing networks than the existing ones.

Keywords: Quantum Computation, Reversible Computation, Quantum Logic Circuit, Excess-3 Code

Abbreviations: BCD - binary coded decimal

1. INTRODUCTION

The industry of integrated circuits has been rapidly developing up to now according to the Moore's law [1]. Thanks to such development, the size of electronic components became smaller and smaller towards an atomic scale. It may be however impossible to miniaturize electronic components below a certain limit. This fact may lead to unsustainability of classical electronic science. Then, the classical electronic circuits would be naturally transferred to quantum based circuits composed of quantum devices, in which arithmetic operation conforms to quantum mechanics. Quantum computing algorithms are much powerful than existing ones which operate on the basis of classical mechanics. One of the merits of quantum computation is that it employs the mechanism of reversible computation, while the operation in classical computers is in general irreversible. During the operation of classical computers, a significant amount of heat is released due to the irreversibility of the operating process. Notice that the operation of most classical gates, such as OR, EXOR, and NAND, is irreversible. In particular, as the circuits of a classical computer become complicated, they release more heat. However, quantum computation with reversible logic gates is free from such kind of heat release.

For the loss of information that corresponds to a single bit in a classical computer that operates at an absolute temperature T , the amount of heat released is $kT \ln 2$ (k : Boltzmann constant). This is known as the Landauer's principle [2]. If a complete history of each non-reversible operation is maintained for any computational algorithm, the computational process is logically reversible. In case that the process of computation is reversible, the elements of a computer do not undergo energy dissipation related to the information loss. Hence, reversible computation [3-10] is important for next generation technologies of computing systems. Bennett examined the Landauer's principle and showed that the design of a reversible algorithm for the general-purpose of a quantum computer is possible [5]. There are several technologies for implementing reversible quantum logic circuits such as quantum-optical networks [11], optical computing [12], programmable logic circuit array [8], etc. If the size of the elements of a circuit becomes comparable to an atomic one, quantum effects would be conspicuous [13]. The basic code used for computation is binary code. However, during the conversion of binary code into decimal one or vice versa, errors occur because there is no one-to-one correspondence between binary and decimal codes [9]. While a decimal code is necessary, binary code is inadequate in order to keep the precision of computing results. The most fundamental decimal code for computation is BCD (binary coded decimal) code [14]. Another useful decimal code is Excess-3 code. Excess-3 code would be convenient [3,15-17] in implementing computations thanks to its particular feature which is the property of self-complement. This is the merit of the use of Excess-3 code, which facilitates the design of logic circuits using Full and Half Adders [15]. In the previous paper [3], we had designed an optimal reversible quantum computing system using quantum gates based on Excess-3 code. To execute quantum computations optimally, it may be necessary to minimize quantum resources. This can be fulfilled by the simplification of an operating circuit for computation. The primitive circuit given in Ref. [3] is composed of 22 operation lines and its improvement has been suggested in the same reference. In this work, we will further minimize the element of logic circuit in order to provide a more improved reversible quantum-computing resource that operates efficiently.

2. MATERIALS AND METHODS

We improve a quantum logic circuit for reversible quantum computation designed in Fig. 5 of Ref. [3] in order for realizing efficient quantum computing process based on Excess-3 code. A garbage qubit line in the previous design will be removed and its role will be undertaken by another qubit line. We will test the newly designed circuit whether it works well or not.

3. RESULTS AND DISCUSSION

3.1. Design for improved quantum circuit

We can compose quantum logic circuits for computation using Full and Half Adders introduced in Refs. [3] and [4]. Quantum circuits operate reversibly in general; This aspect is essentially different from that of classical circuits which use non-reversible circuit elements. In a quantum circuit, the allowed binary value of each qubit during its time evolution is 0, 1, and both 0 and 1 at the same time in certain situations. The number of output qubits is exactly the same as the input qubit, because input and output qubits are the same device. Hence, there is a one-to-one correspondence between input and output qubit information. This is the reason why we can perform reversible computation using quantum circuits. On the other hand, the number of output bits is less than the input ones in the classical computation, leading to the dissipation of heat energy during the computation process according to the loss of information.

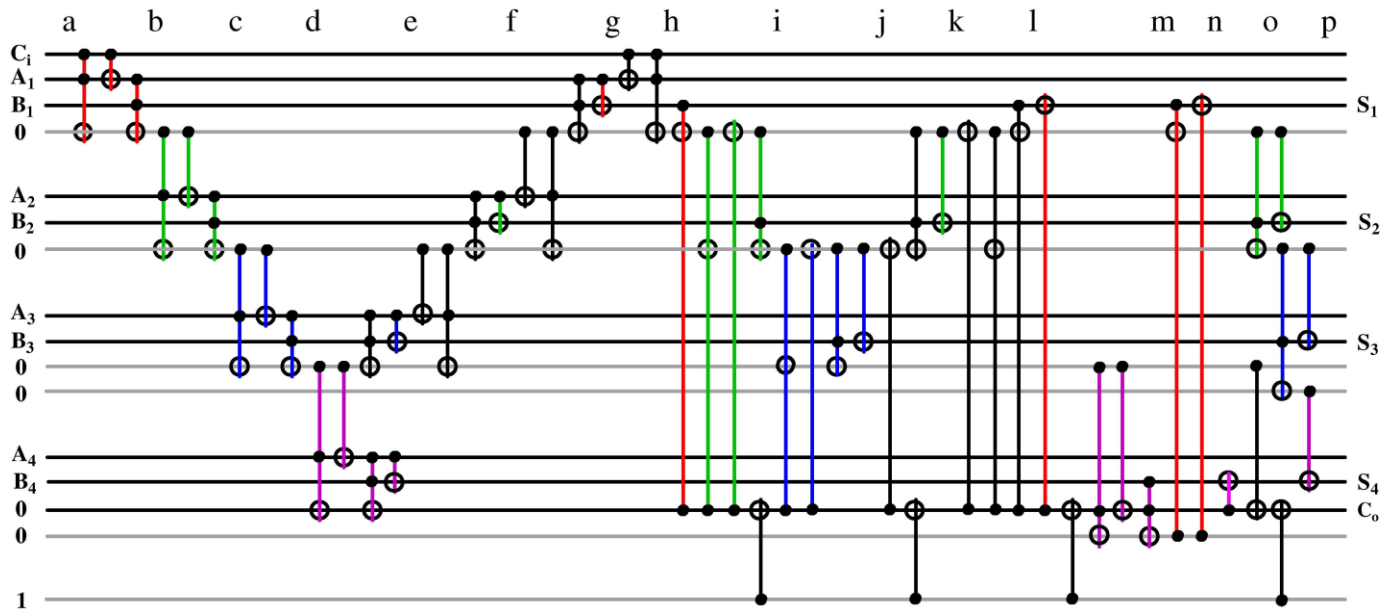


Figure 1 Improved Quantum Circuit that Performs Adding Process by Excess-3 code

In quantum computation, it is important to reduce the size of computing devices through minimization of the number of qubit devices relevant to garbage outputs. This is possible by establishing well organized synthetic methods for device design. For this context, garbage outputs and quantum costs are primary concerns [10].

Our improved design for an optimal logic circuit for a decimal Adder is given in Fig. 1. The process of operation associated with this Adder is the same as that of the previously designed circuit (see Fig. 5 of Ref. [3]) until point i. After point i, the design is improved. Through this improvement, we have reduced a garbage line from the previous one. The number of garbage lines in Fig. 1 is 6 while that in Fig. 5 of Ref. [3] is 7. The simplification of computation by this reduction of a garbage line is our main contribution in this research.

The garbage lines in Fig. 1 are represented in gray colour in order to distinguish them from non-garbage lines. The initial qubit value of the first five garbage lines is 0 and that of the last garbage line is 1. The roles of the zero input garbage line (eighth line in Fig. 5 of Ref. [3]) removed from the circuit of the previous research are undertaken by another garbage qubit line (seventh line in Fig. 1). In order to reuse the garbage line for that purpose, its value has been reset to zero at an appropriate step in the process.

Table 1 Time evolution of qubit values of the circuit given in Fig. 1 for the operation that corresponds to Example 1

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	
C_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	S_1
A_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_1	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	
A_2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	S_2
B_2	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	
	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	
A_3	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	S_3
B_3	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	
	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_4	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	S_4
B_4	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	C_o

3.2. Test for the designed circuit

To see whether our improved circuit works well with the reduced garbage lines, we test the arithmetic computations in this section. For this purpose, let us consider the following two examples of arithmetic computations.

• **Example 1:** At first, we check for the evaluation $4+3=7$ (in Excess-3 code: $0111 + 0110 = 1010$) using Excess-3 code. In this operation, there is no carry output.

• **Example 2:** We consider $8+5=13$ ($1011 + 1000 = 0100\ 0110$) for another operation example. Notice that this process produces a carry 1 as a result.

For these two examples, we assume that there has been no carry received from previous computation process. This means that C_i in Fig. 1 is always zero.

For better understanding of the operation of the circuit, we have represented the evolution of qubit values during the operations for the two cases represented in the above examples. From table 1, we see that the resulting qubit values for example 1 are $(S_4S_3S_2S_1)=(1010)$. This consequence is exact. The output qubit values appearing in table 2, which correspond to the other example, are $(S_4S_3S_2S_1)=(0110)$ with a carry; We can also easily see that this result is correct. Hence, it is obvious that the Excess-3 Adder depicted in Fig. 1 gives exact results. From these checks using tables 1 and 2, we can confirm that the circuit we have proposed in this research works well.

4. CONCLUSION

We have improved a previous decimal computing circuit designed using Excess-3 code. In case that a computation circuit is complicated, errors may take place when we process the computation. If there is a decoherence during the computation, the quantum computing system may undergo change to be a classical-like one which cannot fulfill quantum computing processes. Hence, the decoherence that is apt to occur during a computation is a fatal obstacle in realizing quantum computers. It may be possible to reduce the outbreak of such a decoherence by simplifying arithmetic processes in computing. If we think this fact, the simplification of an operating process for quantum computing is important. For practical circuit design with robust implementation, the running time of a computing process should be shortened.

Table 2 The same as table 1, but for the operation that corresponds to Example 2

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	
C_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	S_1
A_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B_1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	S_2
B_2	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	
A_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	S_3
B_3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	S_4
B_4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	1	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	C_o

While the Excess-3 Adder appeared in the previous work (see Fig. 5 of Ref. [3]) has 17 operation lines and 7 garbage lines, our improved circuit represented in Fig. 1 has 16 operation lines and six garbage lines. Thus, the operation of quantum computing has been simplified in the design in our present work. We have checked, by using two examples of computation processes, that our circuit gives exact results. The advantage of computation using Excess-3 code is based on the fact that the 1's complement of an arbitrary number represented in terms of Excess-3 code is identical to its 9's complement within Excess-3 basis.

DISCLOSURE STATEMENT

This research was supported by the Basic Science Research Program of the year 2016 through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No.: NRF-2016R1D1A1A09919503).

REFERENCE

1. G. E. Moore, Cramming more components onto integrated circuits. *Proc. IEEE*, vol. 86, no. 1, 82-85 (1998).
2. R. Landauer, Irreversibility and heat generation in the computing process. *IBM J. Res. Develop.*, vol. 5, no. 3, 183-191 (1961).
3. J. R. Choi, Optimal logic circuit design for reversible quantum computation based on Excess-3 code. *Discovery*, vol. 52, no. 246, 1483-1493 (2016).
4. K. H. Yeon, J. R. Choi, D. Kim, M.-S. Kim, M. Maamache, Reversible quantum computation using Excess-3 code. *AIP Conf. Proc.*, vol. 1444, 310-313 (2012).
5. C. H. Bennett, Logical reversibility of computation. *IBM J. Res. Develop.*, vol. 17, no. 6, 525-532 (1973).
6. C. H. Bennett, Time/space trade-offs for reversible computation. *Siam J. Comput.*, vol. 18, no. 4, 766-776 (1989).
7. T. Yokoyama, Reversible computation and reversible programming languages. *Electron. Notes Theor. Comput. Sci.*, vol. 253, no. 6, 71-81 (2010).
8. D. Sarvani, Y. Syamala, Y. Ratna Babu, Realization of reversible Full Adder & reversible Full Subtractor using RPLA. *Int. J. Eng. Res. Appl.*, ASPC, 56-59 (2012).
9. B. H. Thapliyal, H. R. Arabnia, R. Bajpai, K. K. Sharma, Partial reversible gates (PRG) for reversible BCD arithmetic. Proceedings of the 2007 International Conference on Computer Design (CDES'07), Las Vegas, U.S.A, June 2007, pp. 90-91 (CSREA Press).
10. M. Mohammadi, M. Eshghi, M. Haghparast, A. Bahrololoom, Design and optimization of reversible BCD Adder/Subtractor circuit for quantum and nanotechnology based systems. *World Appl. Sci. J.*, vol. 4, no. 6, 787-792 (2008).

11. K. Stannigel, P. Rabl, P. Zoller, Driven-dissipative preparation of entangled states in cascaded quantum-optical networks. *New J. Phys.*, vol. 14, no. 6, 063014 (2012).
12. S. Kaur, R. S. Kaler, All optical integrated full adder-subtractor and demultiplexer using SOA-based Mach-Zehnder interferometer. *Int. J. Eng. Sci. Technol.*, vol. 4, no. 1, 303-310 (2012).
13. J. R. Choi, Squeezing effects applied in nonclassical superposition states for quantum nanoelectronic circuits. *Nano Convergence*, vol. 4, 17 (2017).
14. H. Pelka, Computation with BCD (Binary-Coded-Decimal) numbers. *Elektroniker*, vol. 15, el14-el20 (1976).
15. V. V. Blatov, A. A. Chudov, Binary-decimal adder-subtractors. *Instrum. Exp. Tech.*, vol. 21, no. 5 pt 1, 1260-1264 (1978).
16. V. S. P. Nayak, N. Ramchander, R. S. Reddy, T. Marandi, Analysis and design of reversible excess-3 adder and subtractor. IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology, Bangalore, India, May 2016, pp. 397-400.
17. A. Rahman, Md. A. Habib, A. N. Bahar, Z. Rahman, Novel design of BCD to Excess-3 code converter in quantum dots cellular automata (QCA). *Global J. Res. Eng. F: Electr. Electron. Eng.*, vol. 14, no. 4, 7-11 (2014).